Configuring Security (J2EE only)

The Natural Web I/O Interface client and Natural for Ajax come as standard J2EE applications. For the ease of installation, the access to these applications is by default not secured. You might, however, wish to restrict the access to certain parts of these applications to certain users. An important example is the configuration tool, which enables you to modify the Natural session definitions and the logging configuration of the Natural Web I/O Interface client and of Natural for Ajax. Other examples are the Application Designer development workplace contained in Natural for Ajax or the Natural logon page.

This chapter does not cover the concepts of J2EE security in full extent. It provides, however, sufficient information to activate the preconfigured security settings of the Natural Web I/O Interface client and of Natural for Ajax and to adapt them to your requirements. More information on the topics described in this chapter can be found, for instance, at *http://www.jboss.org/jbossas/docs/* (security on JBoss is described in the *Configuration Guide*) or *http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html* (see the chapter on security).

This chapter covers the following topics:

- Name and Location of the Configuration File
- Activating Security
- Defining Security Constraints
- Defining Roles
- Selecting the Authentication Method
- Choosing the Login Module (JBoss Application Server only)
- Defining the Security Realm and Users (Sun Java System Application Server only)

Name and Location of the Configuration File

Security is configured in the file *web.xml*. The path to this file depends on the application server and the type of client that you are using.

- **JBoss Application Server**
  Natural Web I/O Interface client:

  <application-server-install-dir>server/default/deploy/natuniapp.ear/natuniweb.war/WEB-INF

  Natural for Ajax:

  <application-server-install-dir>/server/default/deploy/njx<nn>.ear/cisnatural.war/WEB-INF

- **Sun Java System Application Server**
  Natural Web I/O Interface client:

<application-server-install-dir>/domains/domain1/applications/j2ee-apps/natuniapp/natuniweb_war/WEB-INF

Natural for Ajax:

<application-server-install-dir>/server/domains/domain1/applications/j2ee-apps/njx<nn>/cisnatural_war/WEB-INF

Activating Security

Great care must be taken when editing and changing the configuration file *web.xml*. After a change, the application server must be restarted.

Edit the file *web.xml* and look for the section that is commented with "Uncomment the next lines to add security constraints and roles.". Uncomment this section by removing the comment marks shown in boldface below:

```
<!-- Uncomment the next lines to add security constraints and roles. -->
<!--
<security-constraint>
    <web-resource-collection>
    <web-resource-name>Configuration Tool</web-resource-name>
        <url-pattern>/conf_index.jsp</url-pattern>
        <url-pattern>/faces/*</url-pattern>
    </web-resource-collection>
...
<security-role>
    <description>Administrator</description>
    <role-name>nwoadmin</role-name>
</security-role>
-->
```

Defining Security Constraints

The security constraints defined by default are just examples. A `<security-constraint>` element contains of a number of `<web-resource-collection>` elements combined with an `<auth-constraint>` element. The `<auth-constraint>` element contains a `<role-name>`. The whole `<security-constraint>` element describes which roles have access to the specified resources.

Example - the following definition specifies that only users in the role "nwoadmin" have access to the configuration tool:

```
<security-constraint>
    <web-resource-collection>
    <web-resource-name>Configuration Tool</web-resource-name>
        <url-pattern>/conf_index.jsp</url-pattern>
        <url-pattern>/faces/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>nwoadmin</role-name>
    </auth-constraint>
</security-constraint>
```

In the following section, you will see where and how the roles are defined.

Defining Roles

A few lines below in the file *web.xml*, there is a section `<security-role>`. Here, the roles that can be used in `<security-constraint>` elements are defined. You can define additional roles as needed. The assignment of users to roles is done outside this file and will often be done in a user management that is already established at your site.

Example:

```
<security-role>
    <description>Administrator</description>
    <role-name>nwoadmin</role-name>
</security-role>
```

Selecting the Authentication Method

In the file *web.xml*, there is a section `<login-config>`. The only element that should possibly be adapted here is `<auth-method>`. You can choose between the authentication methods "FORM" and "BASIC". Form-based authentication displays a specific page on which users who try to access a restricted resource can authenticate themselves. Basic authentication advises the web browser to retrieve the user credentials with its own dialog box.

Example:

```
<login-config>
    <auth-method>FORM</auth-method>
...
</login-config>
```

Choosing the Login Module (JBoss Application Server only)

In the directory *<application-server-install-dir>/server/default/conf*, there is a file named *njxnwo-login-config.xml*. The relevant part in this file is the selection of the login module specified in the `<login-module>` element and the configuration of this login module. The login module determines where the user definitions and the assignment of users to roles is maintained.

By default, the `UsersRolesLoginModule` is preconfigured. The `UsersRolesLoginModule` expects the role definitions in one file (*props/njxnwo-roles.properties*) and the user definitions (password and assignment to roles in another file (*props/njxnwo-users.properties*). An example user "admin" with the password "adminadmin" and the role "nwoadmin" is defined to begin with.

You can choose and configure a different login module, for instance, one that expects the user and role definitions in a database or in an LDAP directory, or even write a custom login module.

More information on using JBoss login modules is provided at *http://www.jboss.org/jbossas/docs/* (see the *Configuration Guide*).

Defining the Security Realm and Users (Sun Java System Application Server only)

The following information applies to Sun Java System Application Server 9.1, however, the procedure is similar in other versions.

To create a new security realm and define the user

1. Open the tree node **Configuration** > **Security** > **Realms**.
2. Choose **New**.
3. Enter "NaturalWebIOAndAjaxRealm" as the name of the new realm.
4. Select `com.sun.enterprise.security.auth.realm.file.FileRealm` as the class name.

   Use the following properties which are predefined for this class:

Option	Value
JAAS Context	fileRealm
Key File	${com.sun.aas.instanceRoot}/config/keyfile

5. Choose **OK**.
6. Edit the new realm `NaturalWebIOAndAjaxRealm` and choose the **Manage Users** button.
7. Choose **New**.
8. Enter the user names and the passwords for the users. The name of the group list must be "nwoadmin".
9. Choose **OK**.